MAC239

Lógica de Predicados
(sintaxe)

Agenda

- Motivação:
  - necessidade de uma linguagem mais expressiva
- Lógica de Predicados como uma linguagem formal
  - Termos (variáveis, funções)
  - Fórmulas (predicados, quantificadores)
  - Variáveis livres e ligadas
- Teoria de Prova da Lógica de Predicados
  - Regras de dedução natural

MOTIVAÇÃO

Lógica Proposicional

– Suponha que queremos representar o fato que

"Toda pessoa que está na chuva fica molhada."

$$P \wedge C \rightarrow M$$

para depois usar esse fato. Por exemplo, suponha que nós também sabemos que

"Pedro é uma pessoa e está na chuva."

$$P \wedge C$$

e gostariamos de concluir que "*Pedro ficará molhado*."

– Então, dado $P \wedge C$, podemos de fato concluir M.

A Lógica proposicional não é suficientemente expressiva

- Agora suponha que nos contaram que *"Larissa está na chuva."*

- Gostariamos de concluir que *"Larissa ficará molhada"*, porém …
  - … nada do que declaramos antes nos ajuda a fazer isso!
- O problema é que não somos capazes de representar os detalhes dentro dessas proposições:
  - O raciocínio válido (que nós humanos fazemos) é feito pela **estrutura interna** dessas proposições.
  - Mas na Lógica Proposicional, nós não temos nada além de proposições!

$\Rightarrow$ Precisamos de uma lógica mais expressiva

Lógica de Predicados, também chamada de
Lógica de Primeira Ordem (LPO), do inglês, ***First-order logic*** **(FOL).**

Precisamos de uma lógica mais expressiva

Lógica Proposicional

- lida bem com elementos de sentenças: *não, e, ou, se … então, …*
- mas é limitada para lidar com elementos modificadores: *existe, para todo, somente …*

Exemplo: "***Todo aluno é mais jovem que algum instrutor***"

- Podemos identificar a frase inteira com o símbolo proposicional *p*.
- No entanto, isso não revela a estrutura interna da frase que declara as seguintes propriedades:
  - *ser um aluno*
  - *ser um instrutor*
  - *ser mais jovem que alguém*

Essas são ***propriedades*** de elementos de um conjunto de objeto, que . expressamos na lógica de predicados usando ***predicados.***

Lógica de Predicados

Sintaxe

Predicados, Variáveis e Quantificadores

Propriedades são expressas por ***predicados*** sobre indivíduos: *A, I, J*

A(carlos): Carlos é um aluno.

I(paulo): Paulo é um instrutor.

J(carlos, paulo): Carlos é mais jovem que Paulo.

Variáveis podem ser usadas no lugar dos valores concretos:

A(x): x é um aluno.

I(x): x é um instrutor.

J(x,y): x é mais jovem que y.

- Como expressar: "***Todo aluno***" ?

  \- Precisamos de ***variáveis*** que possam ficar no lugar de valores constantes, e um símbolo de ***quantificador*** que denote "***Todo***" .

- Como expressar: "***algum instrutor***" ?

  \- Precisamos de um símbolo de ***quantificador*** que denote "***Algum***" .

Predicados, Variáveis e Quantificadores

Quantificadores permitem modificar a abrangência dos predicados e representar formalmente a frase:

"Todo aluno é mais jovem que algum instrutor"

Dois quantificadores: Universal (∀) e Existential (∃)

∀x : *para todo* x e

∃y : *existe um* y (ou *existe algum* y)

$$\forall x(A(x) \rightarrow (\exists y(I(y) \wedge J(x,y))))$$

"Para todo x, se x é um aluno então existe algum y tal que y é um instrutor e x é mais jovem que y"

Exemplo: argumento e função

"Nenhum livro é gasoso. Dicionários são livros. Portanto, nenhum dicionário é gasoso."

Denotamos por:

$B(x)$: x é um livro

$G(x)$: x é gasoso

$D(x)$: x é um dicionário

$$\neg \exists x\, (B(x) \wedge G(x)),\ \forall x\, (D(x) \rightarrow B(x)) \vdash \neg \exists x (D(x) \wedge G(x))$$

Exemplo: argumento e função

"Todo filho é mais jovem que sua mãe."

Denotamos por:

$F(x)$: x é um filho
$M(y,x)$: y é mãe de x
$J(x,y)$: x é mais jovem que y

$$\forall x\ \forall y\ (F(x) \wedge M(y,x) \rightarrow J(x,y))$$

Note que y foi usado para denotar a mãe de x.

Podemos representar a frase acima de uma outra forma, definindo a função $m(x)$: mãe de x

$$\forall x\ (F(x) \rightarrow J(x,m(x)))$$

Usar a função $m(x)$ para denotar "*mãe de*" é mais adequado, uma vez que toda pessoa tem apenas uma mãe.

Exemplo: igualdade

"*Carlos e Paulo tem a mesma avó materna.*"

$$\forall x\ \forall y\ \forall u\ \forall v\ (M(x,y) \wedge M(y,carlos) \wedge M(u,v) \wedge M(v,paulo) \rightarrow x=u)$$

Introduzimos um predicado especial: *igualdade.*

Na representação alternativa com funções, teriamos:

$$m(m(carlos) = m(m(paulo)$$

Obs: Nem todo predicado pode ser transformado numa função. Por exemplo, a relação x é irmão de y , deve ser codificada como um predicado, $B(x,y)$, uma vez que uma pessoa pode ter mais do que um irmão.

Predicados

Def1. (Relação). Seja $D_1, D_2, \ldots D_n$ conjuntos, não necessariamente disjuntos. R é uma relação de aridade n sobre os conjuntos $D_1, D_2, \ldots D_n$ se R é um subconjunto do produto cartesiano $D_1 \times D_2 \times \cdots \times D_n$.

Def2. (Predicado da LPO). Seja $D_1, D_2, \ldots D_n$ conjuntos, e R uma relação de aridade n sobre os conjuntos $D_1, D_2, \ldots D_n$. O predicado P associado à R é a ***função total*** de $D_1 \times D_2 \times \cdots \times D_n$ para $\{T, F\}$, ou seja,

$\mathrm{P}: D_1 \times D_2 \times \cdots \times D_n \rightarrow \{T, F\}$, sendo $\mathrm{P}(X) = T$ sse $X \in R$.

Na LPO, consideramos um único conjunto de objetos de discurso D, ou seja $D_1, D_2, \ldots D_n$ são idênticos.

Um predicado nada mais é que uma proposição paramétrica, cujo valor é verdadeiro para alguns elementos de um determinado conjunto, e falso para os demais.

.

Funções da LPO

Def2. (Função da LPO). Seja D o conjunto de objetos de discurso, e $\mathbb{R}$ uma relação de aridade n sobre D. A função f associada à $\mathbb{R}$ é a ***função total*** de D^n para D, ou seja, $f: D^n \rightarrow D$.

.

Lógica de Predicados como uma linguagem formal

O vocabulário define o repertório de símbolos da linguagem da lógica de predicados:

– $\mathcal{P}$ = {p, q, r, ...} é um conjunto de *símbolos de predicados* (cada um com uma aridade fixa);
– $\mathcal{F}$ = {a, a1, b, ... , f, f', g, ...} é um conjunto de *símbolos de funções* (cada um com sua aridade);
– $\mathcal{C} = \{c_1, c_2, ...\}$ é um conjunto de símbolos chamados de constantes (também visto como uma função de aridade 0); e
– $\mathcal{X} = \{x, x_1, ..., x', ...y, z, ...\}$ é um conjunto de símbolos chamados de variáveis.

Existem dois tipos de elementos em uma fórmula de predicados:

– **Objetos** tais como *carlos* (Carlos) e *paulo* (Paulo)., sendo que símbolos de funções também se referem a objetos. Objetos e funções são modelados por ***termos****.*
– **Expressões** para as quais podemos atribuir valores verdade. Expressões são modeladas por ***fórmulas***.

Termos

Termos são definidos como:

- Qualquer variável $x \in \mathcal{X}$ é um termo;
- Qualquer constante em $c \in C$ é um termo;
- Se $t_1, \ldots, t_n$, *são termos e* $f \in \mathcal{F}$ tem aridade n, então $f(t_1, \ldots, t_n)$ é um termo;
- Nada mais é um termo.

BNF:

$$t ::= x \mid c \mid f(t, \ldots, t)$$

onde x representa variáveis em $\mathcal{X}$; c representa constantes em C, e f representa uma função com aridade n.

Exemplo: Suponha que n, f e g são símbolos de função de aridade respetivamente igual a 0, 1 e 2. Então g(f(n), n) e f(g(n), f(n))) são termos, mas g(n) e f(f(n), n) não são termos por violarem as aridades dos símbolos.

Exemplo de termo

Se 0, 1 e 2 são constantes, s é uma função unária, e +, -, e * são funções binários, então a expressão:

$$*(-(2,+(s(x),y)),x)$$

é um termo. Podemos usar a notação infixa para funções:

$$(2 - (s(x) + y)) * x$$

Formulas

Definimos o conjunto de *fórmulas* sobre $(\mathcal{P}, \mathcal{F}, \mathcal{C}, \mathcal{X})$ indutivamente, usando a definição anterior de termos.

–Se P é um predicado com $n \geq 1$ argumentos, e $t_1, \ldots, t_n$, são termos sobre $\mathcal{F}$, então $P(t_1, \ldots, t_n)$ é uma fórmula (denotada por *fórmula atômica*).

–Se Φ é uma fórmula então $\neg\Phi$ também é uma fórmula.

–Se Φ e Ψ são fórmulas, então $\Phi \wedge \Psi$, $\Phi \vee \Psi$, $\Phi \rightarrow \Psi$ também são fórmulas.

–Se Φ e Ψ são fórmulas e x é uma variável, então $\forall x\Phi$ e $\exists x\Phi$ também são fórmulas.

BNF:

$$\Phi ::= P(t_1, \ldots, t_n) \mid (\neg\Phi) \mid (\Phi \wedge \Psi) \mid (\Phi \vee \Psi) \mid (\Phi \rightarrow \Phi) \mid \forall x\Phi \mid \exists x\Phi$$

onde P é um predicado de aridade n; t_i são termos, $i \in \{1, \ldots, n\}$, x é uma variável.

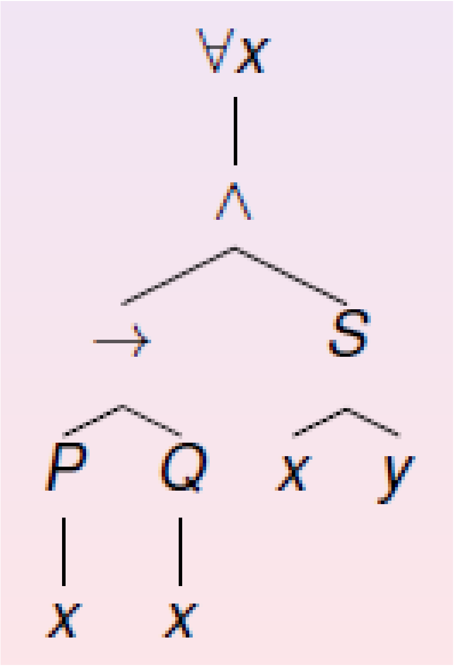
Convenção: Adotaremos a seguinte prioridade entre os operadores:

1. $\neg$, $\forall$, $\exists$
2. $\wedge$, $\vee$
3. $\rightarrow$

Árvore de análise

$$\forall x\ ((P(x) \rightarrow Q(x)) \wedge S(x,y))$$

Tem a árvore de análise (parse)



Exemplo: equivalências

Exemplo 8. Considere a sentença:

"*Nem todos os pássaros podem voar*."

Escolhemos os seguintes predicados para expressar esta sentença:

passaro(x): x é um pássaro .
voa(x): x pode voar.

Esta sentença pode ser codificada da seguinte forma:

$$\neg(\forall x\ (passaro(x) \rightarrow voa(x)))$$

Exemplo: equivalências

Uma outra maneira de expressar a mesma idéia da sentença anterior é dizer que:

" *Existem alguns pássaros que não podem voar*. "

$$\exists x\ (passaro(x) \rightarrow \neg voa(x)))$$

Veremos que as duas codificações são semanticamente equivalentes. De fato, existem transformações que convertem uma na outra.

$$\neg(\forall x\ (passaro(x) \rightarrow voa(x))) \quad \equiv \quad \exists x\ (passaro(x) \rightarrow \neg voa(x)))$$

Exemplo: uso de funções

Como expressar a sentença:

"*Todo filho de meu pai é meu irmão.*"

Duas alternativas:

I.) "*Pai de*" é codificada codificado como um predicado

$F(x,y)$: x é filho de y

$P(x,y)$: x é o pai de y

$B(x,y)$: x é irmão de y

me: *constante*

Tradução: $\forall x \; \forall y \; (P(x,me) \wedge F(y,x) \rightarrow B(y,me))$

II.) "*Pai de*" codificada como uma função

Tradução: $\forall x \; (F(x,p(me)) \rightarrow B(x,me))$

Escopo das variáveis: como em linguagens de programação, as variáveis da LPO possuem um escopo determinado pelos quantificadores.

Abrangência dos quantificadores

Definição: A abrangência de $\forall x\ \Phi$ (ou $\exists x\ \Phi$) é Φ. Uma ocorrência de uma **variável ligada ou presa** (*bound*) numa fórmula Φ é uma ocorrência de uma variável x dentro do campo de abrangência de um quantificador $\forall x$ ou $\exists x$. Uma ocorrência de uma **variável livre** é uma ocorrência de uma variável x não ligada.

Exemplo 1: $\exists x\ (p(f(x),y) \rightarrow q(x))$

As 2 ocorrências da variável x são ligadas, enquanto a ocorrência da variável y é livre.

Exemplo 2: $\exists x p(f(x),y) \rightarrow q(x)$

A 1ª ocorrência da variável x é ligada, enquanto a 2ª é livre.

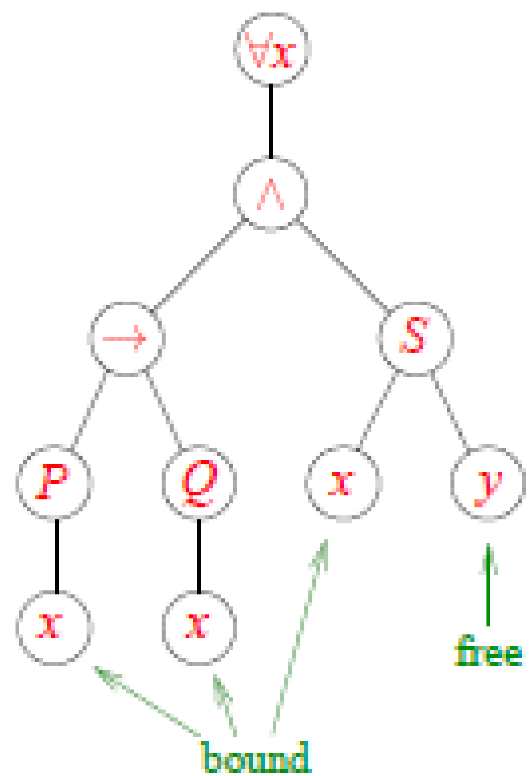
Variáveis livres e variáveis ligadas

Definição: Seja Φ uma fórmula na lógica de predicados. Uma ocorrência de x *é livre em* Φ se ela é um nó folha na árvore de análise de Φ tal que não há um caminho que vai do nó x para um nó $\forall x\Phi$ ou $\exists x\Phi$. Caso contrário, ela é chamada de *ligada* (*bound*).

Fórmula:

$$\forall x\ \underbrace{((P(x) \rightarrow Q(x)) \wedge S(x,y))}_{\text{Escopo de } \forall x\Phi}$$

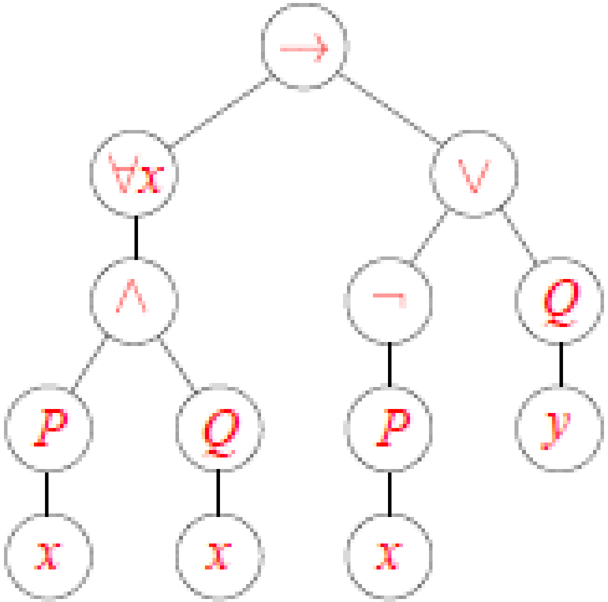
x é ligada e y é livre



Exemplo de uma variável livre e ligada

Fórmula: $(\forall x\ (P(x) \wedge Q(x))) \rightarrow (\neg P(x) \vee Q(y))$

Árvore de análise:



As 2 primeiras ocorrências de x são ligadas; a 3ª ocorrência de x é livre.

Variáveis livres e variáveis ligadas

- Uma fórmula sem variáveis livres é chamada de fórmula fechada (*closed formula*)

- Exemplo: $\forall x \forall y (P(f(x)) \rightarrow \neg(P(x)) \rightarrow Q(f(y), x)))$

Substituição

Variáveis podem ser substituídas por termos: isso é fundamental para a aplicação de regras de inferência.

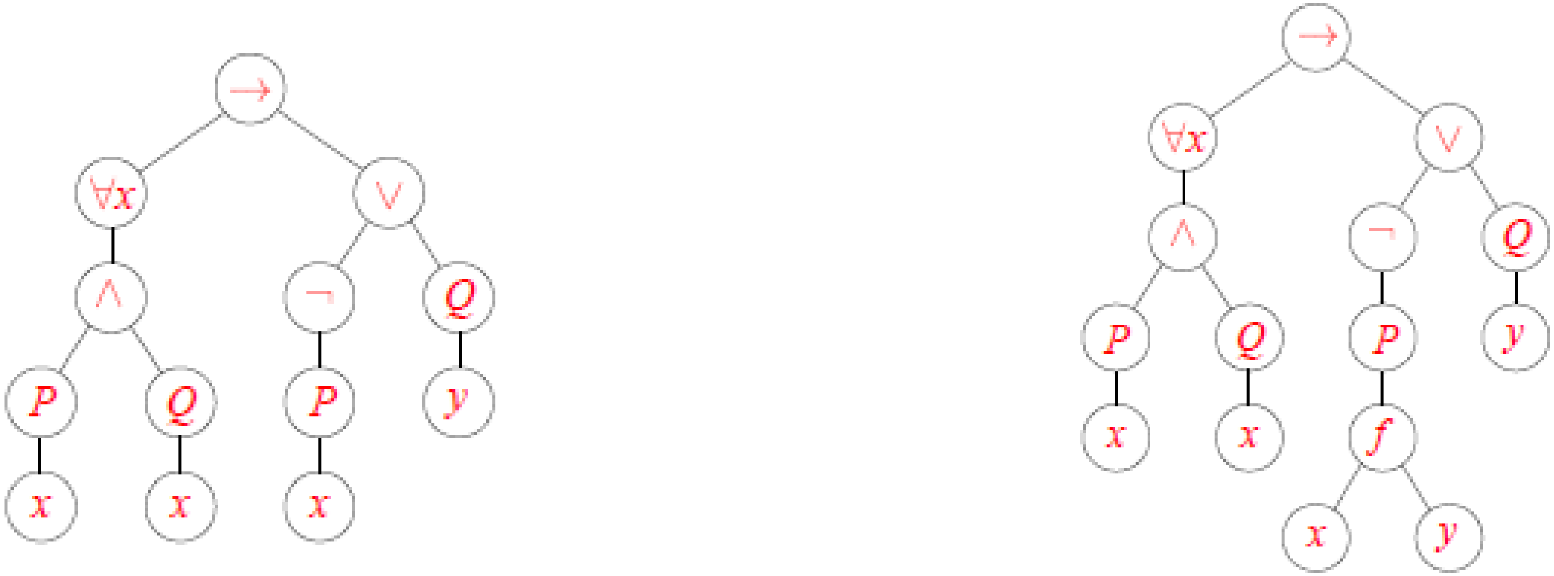
Definição: Dada uma variável *x*, um termo *t* e uma fórmula Φ, definimos Φ[t/x] como a fórmula obtida após substituir cada ocorrência livre da variável x em Φ por t.

Por exemplo, considere a fórmula Φ :

$$(\forall x\ (P(x) \wedge Q(x))) \rightarrow (\neg P(x) \vee Q(y))$$

Nesse caso, temos uma ocorrência livre de x e portanto Φ[f(x,y)/x] resulta em:

$$(\forall x\ (P(x) \wedge Q(x))) \rightarrow (\neg P(f(x,y)) \vee Q(y))$$



Substituição (2)

Exemplo:

$$\forall x \, ((P(x) \rightarrow Q(x)) \wedge S(x,y))$$

Considere a substituição $\Phi[f(x,y)/x]$

Observe que, como todas as ocorrências de x são ligadas, essa substituição não tem nenhum efeito em Φ.

Substituição (3)

As substituições podem produzir efeitos indesejados. Considere o termo $f(x,y)$ e a fórmula $\Phi \equiv \forall y\ (P(x,y))$. Então $\Phi[f(x,y)/x]$ resulta na fórmula:

$$\forall y\ (P(f(x,y),y)).$$

Observe que o termo resultante possui uma semântica diferente da esperada porque a variável y do termo $f(x,y)$ não corresponde à variável y quantificada universalmente na fórmula dada. Como resolver este problema?

Substituição (4)

Definição: Dada uma variável x, um termo t e uma fórmula Φ, dizemos que *t é livre para x em Φ* se nenhuma ocorrência livre de x está no escopo de $\forall y$ ou $\exists y$ para qualquer variável y que ocorra em t.

Exemplo: Considere a fórmula $S(x) \wedge \forall y(P(x) \rightarrow Q(y))$, que possui duas ocorrências livres de x. A ocorrência de x mais a esquerda poderia, por exemplo, ser substituída pelo termo $f(x,y)$, no entanto a outra ocorrência não poderia ser substituída por este termo porque tal substituição acarretaria *captura* da variável y (isto é, o termo $t=f(x,y)$ NÃO é livre para x em Φ.)

Quando precisamos realizar uma substituição de um termo t que não está livre para uma variável x em uma fórmula Φ, o que fazemos é renomear as variáveis ligadas para evitar capturas.

Substituição (5)

No caso do exemplo anterior, a substituição de x pelo termo $f(x,y)$, em

$$S(x) \land \forall y(P(x) \rightarrow Q(y)),$$

pode ser resolvida renomeando a variável ligada y da fórmula para algum nome novo, por exemplo w, tornando a fórmula:

$$S(x) \land \forall w(P(x) \rightarrow Q(w)).$$

Agora a substituição pode ser realizada sem provocar *captura de variáveis* no escopo de quantificadores:

$$S(x) \land \forall w(P(f(x,y)) \rightarrow Q(w))$$

BNF para sentenças da FOL (resumo)

Φ := <Sentence>

<Sentence> := <AtomicSentence>

| <Sentence> <Connective> <Sentence>

| <Quantifier> <Variable>,... <Sentence>

| ¬ <Sentence>

| (<Sentence>)

<AtomicSentence> := <Predicate> (<Term>, ...)

<Term> := <Function> (<Term>, ...)

| <Constant>

| <Variable>

<Connective> := ∧ | v | → | ↔

<Quantifier> := ∃ | ∀

<Constant> := “c" | “x1" | “john" | ...

<Variable> := "a" | "x" | "s" | ...

<Predicate> := “before" | “hasColor" | “raining" | ...

<Function> := “mother" | “leftLegOf" | ...

Próxima aula

- Teoria de Prova para LPO
  - Regras de inferência
  - Exemplos de provas de argumentos